

Grid refinement in the construction of Lyapunov functions using radial basis functions

Article (Accepted Version)

Giesl, Peter and Mohammed, Najla (2015) Grid refinement in the construction of Lyapunov functions using radial basis functions. *Discrete and Continuous Dynamical Systems - Series B*, 20 (8). pp. 2453-2476. ISSN 1531-3492

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/57408/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

Copyright and reuse:

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

GRID REFINEMENT IN THE CONSTRUCTION OF LYAPUNOV FUNCTIONS USING RADIAL BASIS FUNCTIONS

NAJLA MOHAMMED, PETER GIESL

Department of Mathematics
University of Sussex
Falmer BN1 9QH, United Kingdom

ABSTRACT. Lyapunov functions are a main tool to determine the domain of attraction of equilibria in dynamical systems. Recently, several methods have been presented to construct a Lyapunov function for a given system. In this paper, we improve the construction method for Lyapunov functions using Radial Basis Functions. We combine this method with a new grid refinement algorithm based on Voronoi diagrams. Starting with a coarse grid and applying the refinement algorithm, we thus manage to reduce the number of data points needed to construct Lyapunov functions. Finally, we give numerical examples to illustrate our algorithms.

1. Introduction. The determination of the domain of attraction of an equilibrium is an important task in the analysis and derivation of dynamical systems, arising in many practical applications. Since it is difficult to determine the exact domain of attraction analytically, researchers have been seeking numerical algorithms to determine subsets of the domain of attraction. Most of these methods for computing domains of attraction are based on Lyapunov functions, which are functions that decrease along trajectories of the dynamical system. Sublevel sets of Lyapunov functions are positively invariant subsets of the domain of attraction. The construction of such Lyapunov functions, however, is very challenging.

In the last decades, several numerical methods to construct Lyapunov functions have been developed, for a review see [9]. These methods include the **SOS** (sums of squares) method, which is applicable for polynomial vector fields, introduced in [23] and available as a MATLAB tool box [22]. It constructs a polynomial Lyapunov function by **semidefinite optimization**.

The **CPA** method constructs a CPA (continuous piecewise affine) Lyapunov function using **linear optimization** [13, 14]. A simplicial complex is fixed and the space of CPA functions which are affine on each simplex is considered. This space can be parameterized by the values on the vertices. The conditions of a Lyapunov function are transformed into a set of finitely many linear inequalities at the vertices, which include error estimates ensuring that the CPA Lyapunov function has negative orbital derivative inside each simplex. These linear inequalities are used as constraints of a linear programming problem, which can be solved by standard methods. While in the original method an arbitrarily small neighborhood of the equilibrium had to be cut out, a revised method can construct a CPA Lyapunov

Key words and phrases. Lyapunov function, domain of attraction, mesh-free collocation, Radial Basis Function, Voronoi diagram, refinement.

function also near the equilibrium by using a fan-like triangulation near the equilibrium [8].

A different method deals with **Zubov's equation** and computes a solution of this partial differential equation (PDE) [3]; the corresponding generalized Zubov equation is a Hamilton-Jacobi-Bellmann equation. This equation has a viscosity solution which can be approximated using standard techniques after regularisation at the equilibrium, for example one can use piecewise affine approximating functions and adaptive grid techniques [11].

The **cell mapping approach** [15] or **set oriented methods** [4] divide the phase space into cells and compute the dynamics between these cells; they have also been used to construct Lyapunov functions [12].

The **RBF** (Radial Basis Function) method, a special case of mesh-free collocation, considers a particular Lyapunov function, satisfying a linear PDE and solves it using **mesh-free collocation** [6, 10]. For this method, a set of scattered grid points is used to find an approximation to the solution of the linear PDE. It is computed by solving a linear system of equations, for more details see Section 3.

In this paper, we extend the Radial Basis Function method to construct Lyapunov functions which will in turn determine subsets of the domain of attraction through sublevel sets of these Lyapunov functions. Our new contribution is to combine this construction method with a grid refinement algorithm in order to reduce effort and computation time.

The RBF method is very well suited for a refinement algorithm as the grid points can be scattered and do not require a special structure. Usually, if too few grid points are used, the constructed function has some areas with positive orbital derivative. Instead of refining the grid uniformly, the idea of the refinement algorithm is to refine the grid only where we need to do so, i.e., in areas with positive orbital derivatives. This is particularly useful when going to higher dimensions to save computation time both when constructing the function and evaluating the function, e.g. to determine sublevel sets.

The outline of the paper will be as follows: Section 2 gives an introduction to dynamical systems and Lyapunov functions. In Section 3 we explain the steps of the RBF construction method in detail. In Section 4 we present our refinement algorithm and apply it to numerical examples. Finally, Section 5 concludes the paper.

2. Dynamical systems and Lyapunov functions. Consider the following autonomous system of differential equations, which describes a dynamical system

$$\dot{x} = f(x) \tag{1}$$

where $f \in C^\sigma(\mathbb{R}^d, \mathbb{R}^d)$, $\sigma \geq 1$ and $d \in \mathbb{N}$. We denote by $S_t \xi := x(t)$ the solution of the initial value problem $\dot{x} = f(x)$, $x(0) = \xi \in \mathbb{R}^d$, $t \geq 0$ and we assume that the solution exists for all positive times. Note that S_t defines a dynamical system on \mathbb{R}^d .

A point $x_0 \in \mathbb{R}^d$ is an equilibrium point of (1) if $f(x_0) = 0$. Then, $x(t) = x_0$ is a constant solution of (1) for all $t \geq 0$. The stability of an equilibrium point is determined by the behaviour of solutions in a neighborhood of an equilibrium. An equilibrium is called *stable*, if all solutions starting near the equilibrium point stay near the equilibrium for all future times. Moreover, it is called *asymptotically stable* if it is stable and the solutions starting near the equilibrium converge to it as

time tends to infinity. It is called *exponentially stable*, if the convergence to the equilibrium is exponential. For an asymptotically stable equilibrium x_0 we are interested in finding the largest set of initial states from which the trajectories of solutions converge to the equilibrium as time tends to infinity. This set is called the *domain of attraction*.

Definition 1 (Domain of attraction). *The domain of attraction of an asymptotically stable equilibrium x_0 is defined by*

$$A(x_0) := \left\{ x \in \mathbb{R}^d \mid S_t x \xrightarrow{t \rightarrow \infty} x_0 \right\}. \quad (2)$$

Remark 1. *The domain of attraction $A(x_0)$ of an asymptotically stable equilibrium x_0 is non-empty and open.*

2.1. Lyapunov Functions. The method of Lyapunov functions enables us to determine subsets of the domain of attraction of an asymptotically stable equilibrium through sublevel sets. A function $V \in C^1(\mathbb{R}^d, \mathbb{R})$ is called a Lyapunov function for the equilibrium x_0 if it has a local minimum at x_0 and a negative orbital derivative in a neighborhood of x_0 .

Definition 2 (Orbital derivative). *The orbital derivative of a function $V \in C^1(\mathbb{R}^d, \mathbb{R})$ with respect to (1) at a point $x \in \mathbb{R}^d$ is defined by*

$$V'(x) = \langle \nabla V(x), f(x) \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product in \mathbb{R}^d .

Remark 2. *The orbital derivative is the derivative along solutions: with the chain rule we have*

$$\left. \frac{d}{dt} V(S_t x) \right|_{t=0} = \langle \nabla V(S_t x), \dot{x} \rangle \Big|_{t=0} = \langle \nabla V(x), f(x) \rangle = V'(x). \quad (3)$$

The following theorem shows how Lyapunov functions are used to find subsets of the domain of attraction; note that the requirement of the local minimum at x_0 is a consequence of the assumptions. The theorem states that sublevel sets of a Lyapunov function are positively invariant subsets of the domain of attraction, see e.g. [6, Theorem 2.24].

Theorem 1. *Let $x_0 \in \mathbb{R}^d$ be an equilibrium, $V \in C^1(\mathbb{R}^d, \mathbb{R})$ and $K \subset \mathbb{R}^d$ be a compact set with neighbourhood B such that $x_0 \in \overset{\circ}{K}$. Moreover, let*

1. $K = \{x \in B \mid V(x) \leq R\}$ for a $R \in \mathbb{R}$, i.e., K is a sublevel set of V .
2. $V'(x) < 0$ for all $x \in K \setminus \{x_0\}$, i.e., V is decreasing along solutions in $K \setminus \{x_0\}$.

Then $K \subset A(x_0)$, K is positively invariant and V is called a Lyapunov function.

2.2. Existence of Lyapunov functions. There are several results on converse theorems, ensuring the existence of Lyapunov functions in different contexts, e.g. already in 1949 [21]; for a review see [19]. These converse theorems, however, use the explicit solution of (1) and are thus often not useful to construct a Lyapunov function explicitly.

We will assume that x_0 is an exponentially stable equilibrium and we will consider two classes of Lyapunov functions such that $V'(x) < 0$ holds for all $x \in A(x_0) \setminus \{x_0\}$. We will characterise them by equations for $V'(x)$; these are linear first-order Partial Differential Equations (PDE) for V . Both functions have the same degree of smoothness as f , i.e. they are C^σ functions.

- The first class are Lyapunov functions T , which satisfy the equation

$$T'(x) = -\bar{c},$$

where $\bar{c} > 0$ is a given constant. Note, however, that the function T is not defined at x_0 and fulfills $\lim_{x \rightarrow x_0} T(x) = -\infty$.

- The second class are Lyapunov functions which satisfy

$$Q'(x) = -\|x - x_0\|^2$$

or a similar right-hand side.

For more details on the existence of the Lyapunov functions T and Q see [6, Theorems 2.38 and 2.46].

In this paper, we will extend the RBF method to construct Lyapunov functions by a refinement algorithm. In the next section we will recall the main ingredients of this method before we introduce the refinement in Section 4.

3. Construction of Lyapunov functions using Radial Basis Functions. The two Lyapunov functions T and Q , introduced in the previous section, satisfy partial differential equations for their orbital derivative. Therefore, their solutions can be approximated using the Radial Basis Functions collocation method. Note that the approximation itself will be a Lyapunov function.

3.1. Numerical solutions for PDE's using the Radial Basis Functions collocation method. Meshless collocation based on Radial Basis Function is an effective tool to solve linear PDE's. It has outstanding properties, such as approximating arbitrarily scattered data in multidimensional space as well as providing high order of accuracy which have made it a preferable method for the numerical solutions of partial differential equations. For a general introduction to Meshless collocation, in particular Radial Basis Functions, see [2, 28]. For the application of RBF to the construction of Lyapunov functions, see [6], where details for the following overview of the method can be found, as well as [10].

A Radial Basis Function is a real-valued function whose value depends only on the distance from the origin i.e., $\Psi(x) = \psi(\|x\|)$, or alternatively from another point called *centre*, i.e., $\Psi(x, x_j) = \psi(\|x - x_j\|)$, where $\|\cdot\|$ denotes the Euclidean norm in \mathbb{R}^d . There is a one-to-one correspondence between a Radial Basis Function, or more generally kernel, and its Reproducing Kernel Hilbert Space (RKHS). The approximate solution of the PDE will be a norm-minimal interpolant in the RKHS; in our brief overview, however, we do not discuss this relation further, the interested reader is referred to [10].

In the following section we will introduce a family of compactly supported Radial Basis Functions that enables us to approximate functions with certain smoothness i.e., not necessarily C^∞ . Note that the corresponding RKHS is a Sobolev space with equivalent norm.

The Wendland functions, introduced by Wendland [27], are compactly supported Radial Basis Functions, which are polynomials on their support.

Definition 3 (Wendland Functions). *Let $l \in \mathbb{N}$, $k \in \mathbb{N}_0$. We define by recursion*

$$\psi_{l,0}(r) = (1 - r)_+^l \tag{4}$$

and

$$\psi_{l,k+1}(r) = \int_r^1 t\psi_{l,k}(t)dt \quad (5)$$

for $r \in \mathbb{R}_0^+$. Here we set $x_+ = x$ for $x \geq 0$ and $x_+ = 0$ for $x < 0$.

If we fix the parameter $l := \lfloor \frac{d}{2} \rfloor + k + 1$ depending on the space dimension d and the parameter k , then the function $\Psi(x) = \psi_{l,k}(c\|x\|)$ with $c > 0$ is a C^{2k} function with compact support. For dimensions $d = 2$ or $d = 3$, we give some Wendland functions in the following table.

k	$\psi_{l,k}$
1	$\psi_{3,1}(cr) = (1 - cr)_+^4(4cr + 1)$
2	$\psi_{4,2}(cr) = (1 - cr)_+^6(35c^2r^2 + 18cr + 3)$
3	$\psi_{5,3}(cr) = (1 - cr)_+^8(32c^3r^3 + 25c^2r^2 + 8cr + 1)$
4	$\psi_{6,4}(cr) = (1 - cr)_+^{10}(429c^4r^4 + 450c^3r^3 + 210c^2r^2 + 50cr + 5)$

TABLE 1. The Wendland functions $\psi_{3,1}(cr)$, $\psi_{4,2}(cr)$, $\psi_{5,3}(cr)$ and $\psi_{6,4}(cr)$ with $l = k + 2$ and a scaling parameter $c > 0$. Note that these functions are the Wendland functions of Definition 3 up to a constant.

Now let us consider a general linear partial differential equation of the form

$$Lu = g \text{ on } \Omega \subset \mathbb{R}^d, \quad (6)$$

where L is a linear differential operator of the form

$$Lu(x) = \sum_{|\alpha| \leq m} c_\alpha(x) D^\alpha u(x). \quad (7)$$

In our case, the differential operator L will be given by the orbital derivative of a function V with respect to system (1), namely

$$LV(x) := \langle \nabla V(x), f(x) \rangle = \sum_{j=1}^n f_j(x) \partial_j V(x) = V'(x) \quad (8)$$

The operator L in (8) is a first order differential operator of the form (7) with $c_{e_j}(x) = f_j(x)$.

Let $X_N = \{x_1, x_2, \dots, x_N\} \subset \Omega$ be a set of N pairwise distinct points which are no equilibria. Define *Dirac's delta-operator* δ by $\delta_{y_0}g(x) = g(y_0)$. Then we have

$$(\delta_{x_k} \circ L)^x V(x) = LV(x_k) = V'(x_k),$$

where the superscript x denotes the application of the operator with respect to the variable x . The approximant $v : \mathbb{R}^d \rightarrow \mathbb{R}$ of V will be given by

$$v(x) = \sum_{k=1}^N \beta_k (\delta_{x_k} \circ L)^y \Psi(x - y) \quad (9)$$

where $\Psi(x)$ is the Radial Basis Function. The coefficients β_k are determined by claiming that the interpolation condition

$$(\delta_{x_j} \circ L)^x V(x) = (\delta_{x_j} \circ L)^x v(x)$$

is satisfied for all grid points $x_j \in X_N$, or in other words that the PDE is satisfied at all points $x_j \in X_N$. This will lead to a linear system for β , i.e., $\mathbf{A}\beta = \alpha$. If the points x_j are pairwise distinct and no equilibria, the symmetric matrix \mathbf{A} is positive definite, so in particular non-singular. Hence, the system has a unique solution β . The interpolation matrix entries of $\mathbf{A} = (a_{jk})_{j,k=1,\dots,N}$ are given by

$$a_{jk} = (\delta_{x_j} \circ L)^x (\delta_{x_k} \circ L)^y \Psi(x - y)$$

and the right-hand side $\alpha = (\alpha_j)_{j=1,\dots,N}$ is given by

$$\alpha_j = (\delta_{x_j} \circ L)^x V(x) = LV(x_j) = V'(x_j)$$

which are chosen to be one of our Lyapunov functions $V'(x_j) = -\bar{c}$ or $V'(x_j) = -\|x_0 - x_j\|^2$.

Finally, we calculate the approximant $v(x)$ and its orbital derivative $v'(x)$, using the following formulas, by evaluating and taking the orbital derivative of (9).

$$v(x) = \sum_{k=1}^N \beta_k \langle x_k - x, f(x_k) \rangle \psi_1(\|x - x_k\|), \quad (10)$$

$$\begin{aligned} v'(x) &= \sum_{k=1}^N \beta_k [\psi_2(\|x - x_k\|) \langle x - x_k, f(x) \rangle \langle x_k - x, f(x_k) \rangle \\ &\quad - \psi_1(\|x - x_k\|) \langle f(x), f(x_k) \rangle], \end{aligned} \quad (11)$$

where ψ_1 and ψ_2 are defined as:

$$\psi_1(r) = \frac{\frac{d}{dr} \psi(r)}{r}, \quad \text{for } r > 0 \quad (12)$$

$$\psi_2(r) = \begin{cases} \frac{\frac{d}{dr} \psi_1(r)}{r} & \text{for } r > 0 \\ 0 & \text{for } r = 0 \end{cases} \quad (13)$$

Note that ψ_1 can be continuously extended to 0.

Remark 3 (The value of \bar{c}). *Changing the value of \bar{c} has the effect of multiplying the solution β of the linear system $\mathbf{A}\beta = \alpha$ by a positive constant. As a consequence also the value of the approximant v and its orbital derivative v' will be multiplied by the same positive constant. This means that the areas of the phase space, where v' is positive, are independent of the value of \bar{c} .*

Indeed, this follows from $\mathbf{A}\beta = \alpha = -\bar{c} = -(1, 1, \dots, 1)^T \bar{c}$, since the interpolation matrix \mathbf{A} is independent of the value of \bar{c} and from formulas (10) and (11).

The following error estimate was given in [10, Corollary 4.11]. Note, that $W_2^\tau(\Omega)$ denotes the usual Sobolev space on $\Omega \subset \mathbb{R}^d$.

Theorem 2. *Denote by k the smoothness index of the compactly supported Wendland function and fix $l := \lfloor \frac{d}{2} \rfloor + k + 1$. Let $k > \frac{1}{2}$ if d is odd or $k > 1$ if d is even. Set $\tau = k + (d + 1)/2$ and $\sigma = \lfloor \tau \rfloor$. Consider the dynamical system defined by (1), where $f \in C^\sigma(\mathbb{R}^d, \mathbb{R}^d)$. Let x_0 be an exponentially stable equilibrium point of (1). Let f be bounded in $A(x_0)$ and denote by $V \in W_2^\tau(A(x_0), \mathbb{R})$ the Lyapunov function satisfying $V'(x) = -\|x - x_0\|^2$.*

The reconstruction v of the Lyapunov function V with respect to the operator (8) and a set $K \subset \Omega := \{x \in A(x_0) \mid V(x) \leq r\} \setminus \{x_0\}$, $r > 0$, satisfies

$$\|v' - V'\|_{L_\infty(\Omega)} \leq Ch^{k-\frac{1}{2}} \|V\|_{W_2^{k+(d+1)/2}(\Omega)} \quad (14)$$

where $h := \sup_{x \in \Omega} \min_{x_j \in X_N} \|x - x_j\|$ denotes the fill distance, i.e. the maximum distance which a point in Ω can have from any point in X_N .

Remark 4. The set K in the theorem above can be any compact subset of $A(x_0)$ as r can be chosen so large that the sublevel set Ω of V satisfies $\Omega \supset K$. A similar statement for the function satisfying $V'(x) = -\bar{c}$ follows also from [10, Corollary 4.11], but with no data sites on the boundary. Note, however, that in this case the function satisfying $V'(x) = -\bar{c}$ is not unique up to a constant, the error estimates on the orbital derivative, however, still hold.

The error estimate (14) implies that the approximant v of the Lyapunov function V is actually a Lyapunov function, i.e. satisfies $v'(x) < 0$, if the grid is dense enough. Let us make this more precise: by choosing the fill distance h so small that $Ch^{k-\frac{1}{2}} \|V\|_{W_2^{k+(d+1)/2}(\Omega)} \leq \epsilon$ for a given $\epsilon > 0$,

1. for $V'(x) = -\|x - x_0\|^2$ we have with $|V'(x) - v'(x)| \leq \epsilon$, hence
$$v'(x) \leq V'(x) + \epsilon = -\|x - x_0\|^2 + \epsilon < 0 \text{ if } \|x - x_0\|^2 > \epsilon. \quad (15)$$

2. for $V'(x) = -\bar{c}$ we have with $|V'(x) - v'(x)| \leq \epsilon$, hence
$$v'(x) \leq V'(x) + \epsilon = -\bar{c} + \epsilon < 0, \text{ if } \epsilon < \bar{c}. \quad (16)$$

Remark 5. In both cases, the approximation may fail to have negative orbital derivative near the equilibrium x_0 ; in the first case the error estimate requires $\|x - x_0\|^2 > \epsilon$, and in the second case the function V is not defined in x_0 . If the equilibrium is exponentially stable, one can use the Lyapunov function of the linearised system, the so-called local Lyapunov function, to deal with this small neighborhood of x_0 , for details see [6], or a modified method, see [7]. In this paper, we will not deal with this local problem in more detail.

The error estimate uses the fill distance as a measure; hence, we are led to choose a uniformly fine grid. In examples, however, it turns out that an approximation with negative orbital derivative can be achieved with fewer points using a non-uniform grid. Moreover, the goal is not necessarily to have a good approximation of V , but to construct a function with negative orbital derivative. For example, when approximating the solution of $V'(x) = -\|x - x_0\|^2$, a larger error is permissible for points far away from the equilibrium.

3.2. Steps of the construction method. The construction method is based on considering the Lyapunov function satisfying the PDE's stated in Section 2.2. We will explain this construction method in an example.

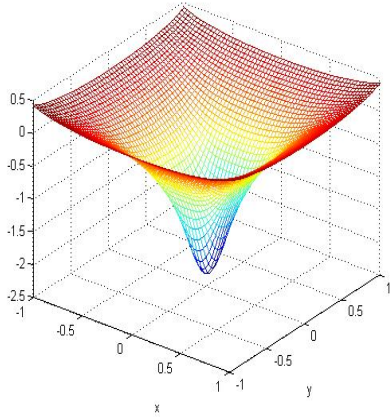
Example 1. Consider the simple linear system

$$\begin{cases} \dot{x} = -x \\ \dot{y} = -y \end{cases}$$

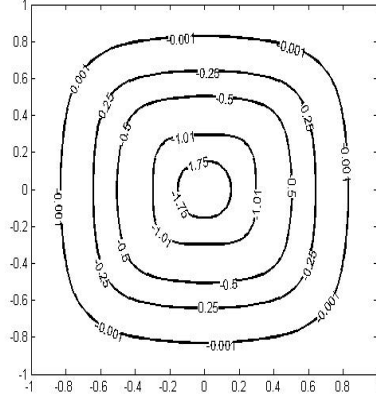
The system has one asymptotically stable equilibrium $x_0 = (0, 0)$. Now we will go through the steps of the method.

- Choose a Radial Basis Function $\Psi(x) = \psi_{l,k}(c\|x\|)$, here we choose the Wendland function $\psi_{6,4}$ with $c = 1$.
- Choose a grid $X_N = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^2$, containing no equilibrium point. Here, we fix a regular grid $X_N = \{(x, y) \in \mathbb{R}^2 \mid x, y \in \{0, \pm h, \pm 2h, \dots, \pm 1\}\} \setminus \{(0, 0)\}$, where $h > 0$ is the distance between points in x - and y -direction and will be specified below.

- Use the RBF method to approximate the Lyapunov function $V = T$ which satisfies $T'(x) = -1$ by the approximant v and then calculate its orbital derivatives v' using (10) and (11).
- From the ansatz of the Lyapunov function V , we know that the orbital derivative is negative at every point in our grid, i.e., $v'(x_i) < 0$ for all $x_i \in X_N \subset \mathbb{R}^2$, but there may be points in $[-1, 1]^2$, where the orbital derivative is positive. The error estimate tells us that if h is small enough, the orbital derivative will be negative except for a small neighborhood of the equilibrium. Hence, we start with a certain h , and check the sign of the orbital derivative at the points between the grid points. If we have points where $v'(x) > 0$, we need a finer grid. Therefore, we choose h smaller and smaller until we have $v'(x) < 0$ for all points in the desired area. In this example, we used $h = 1/6$, resulting in $N = 168$ points.
- Find a sublevel set K of the Lyapunov function (the approximant) v of level $R \in \mathbb{R}$ which is a subset of $\{x \in \mathbb{R}^2 \setminus \{(0, 0)\} \mid v'(x) < 0\}$; then K is a subset of the domain of attraction $A(x_0)$. Figures 1 and 2 show the functions v and v' as well as level sets of v .



(a) The constructed Lyapunov function $v(x, y)$.



(b) Level sets of $v(x, y)$.

FIGURE 1. (a) shows the constructed Lyapunov function v and (b) different sublevel sets of v for different values of R , which are subsets of the domain of attraction.

During the calculations of the Lyapunov function v we may find some points with positive orbital derivatives. This occurs because either the grid we have chosen is not fine enough, or the grid points do not all lie in the domain of attraction. We cannot exclude the second case, so we use a finer grid to calculate the Lyapunov function v and see if the problem will be solved. However, instead of using a regular, finer grid, the question is whether the refinement can be done more efficiently by using an irregular grid with fewer points. Our goal is to have a fine grid but to avoid expensive computations which are caused by refining the whole area rather

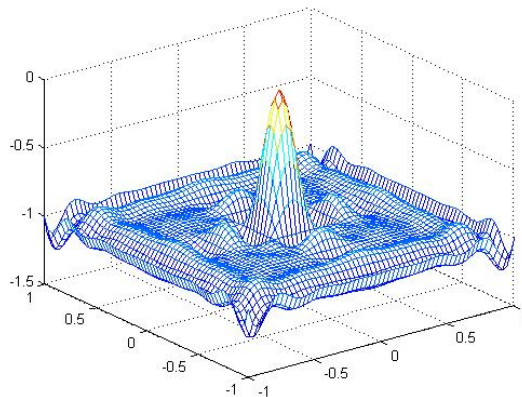


FIGURE 2. The orbital derivative $v'(x, y)$ of the constructed Lyapunov function v ; note that $v'(x) \approx -1$, except for a small neighborhood of the origin.

than only the parts where we need to add more points. Therefore, we combine this construction method with a refinement algorithm.

4. Grid Refinement Algorithm. As the RBF approximation method is mesh-free, i.e. there is no special connectivity between points, the procedure of adding points will be easy and convenient. The proposed algorithm is recursive and uses *Voronoi diagrams*. In each step, given a grid, we generate a Voronoi diagram for our grid points, then we consider the Voronoi vertices of each cell of this diagram as possible points to be added to the grid. Finally, we run a test on each Voronoi vertex and decide whether we add the point to the grid or not.

We have used Voronoi vertices as new possible points for our grid since they are equidistant to three or more previous grid points, and thus lie “in between” the previous grid points. A brief introduction to Voronoi diagram as well as the strategy of the refinement algorithm will be given in the following two sections, respectively.

4.1. Voronoi diagram. A Voronoi diagram is a geometric structure that divides a d -dimensional space into cells based on the distance between sets of points in the space [24]. Many algorithms have been proposed for computing Voronoi diagrams. The fundamental and most popular ones include: The Divide and Conquer algorithm, and Fortunes’s Sweep Line algorithm [1]. For the purpose of explaining the structure of Voronoi diagrams, we will explain a very simple but less efficient algorithm using perpendicular hyperplanes.

Let $S = \{s_1, s_2, \dots, s_n\} \subset \mathbb{R}^d$ be a set of n arbitrarily distributed and distinct sites (points) in \mathbb{R}^d . The perpendicular bisector algorithm works as follows: for each pair of sites in S we construct a perpendicular hyperplane to the line segment joining these sites. At the end of this process, we will have intersections of finitely many hyperplanes which build up cells, with a convex polygon structure, known as *Voronoi regions*. The boundaries of each region are called *Voronoi regions* and the intersections of multiple Voronoi edges are called *Voronoi vertices*. For more details see [1, 20, 17].

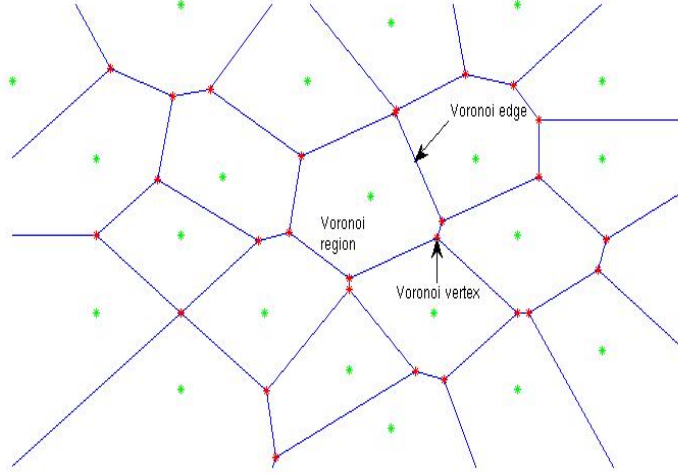


FIGURE 3. The Voronoi diagram for a set of points (green *) with a Voronoi region, Voronoi edge and Voronoi vertex (red *).

Mathematically, the Voronoi region of a point s_i in S is defined by

$$\mathcal{V}_i = \bigcap_{j=1, j \neq i}^n \left\{ x \in \mathbb{R}^d \mid \|x - s_i\| < \|x - s_j\| \right\},$$

where $\|\cdot\|$ denotes the Euclidean distance. This means that for every point $x \in \mathbb{R}^d$ within a Voronoi region \mathcal{V}_i the Euclidean distance of x to the site s_i , which is also inside the region, is smaller than the Euclidean distance of x to any other site s_j .

Remark 6. *As the Voronoi region is the intersection of finitely many hyperplanes, each Voronoi region is a convex polygon.*

Delaunay triangulation is the dual structure of Voronoi diagram. The relation between them is simply that the sites of the Voronoi diagram are the vertices of the Delaunay triangulation and vice versa.

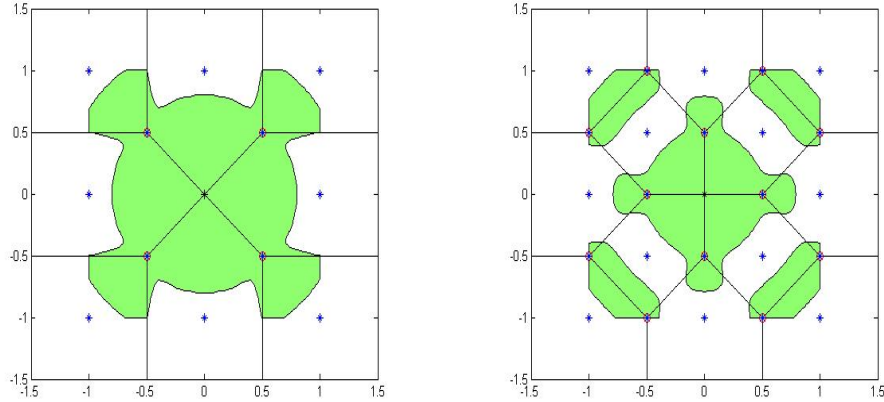
Voronoi diagrams and Delaunay triangulations have enormous applications in different scientific fields, especially in mesh generation and nodes insertion procedures. Sibson [26] developed an interpolation method based on Voronoi diagrams, the method is known as natural neighbour interpolation method. Moreover, there is a refinement algorithm called Ruppert's Delaunay refinement algorithm [25]. Some recent works include [18], presenting a refinement procedure to develop the gradient smoothing method using Delaunay triangulation for the adaptive analysis in solid mechanics, and [29], where a Voronoi neighbour criterion is used to construct the adaptive radial point interpolation method. Voronoi diagrams have also been used in kernel-based adaptive particle methods for numerical flow simulation [16], and for a thinning algorithm in multistep interpolation with Radial Basis Functions [5].

4.2. The refinement algorithm strategy. We will now describe the refinement algorithm in detail.

1. Fix a compact neighbourhood $K \subset \mathbb{R}^d$ of the equilibrium x_0 and a Radial Basis Function. Let $n = 1$ and start with an initial set of grid points $X_n = \{x_1, x_2, \dots, x_{N_n}\} \subset K$, not containing any equilibrium.
2. Calculate a Lyapunov function v_n using the RBF method with the grid X_n .
3. Generate Voronoi vertices, $Y_n = \{y_1, y_2, \dots, y_{M_n}\} \subset \mathbb{R}^d$, for the grid points X_n . Exclude points in Y_n which are equilibria or which lie outside K .
4. Run a test on each vertex in the set Y_n and check whether $v'_n(y_j) < 0$ ($y_j \in Y_n^-$) or $v'_n(y_j) \geq 0$ ($y_j \in Y_n^+$), where $j = 1, \dots, M_n$ and $Y_n = Y_n^- \cup Y_n^+$.
5. Define new grid $X_{n+1} = X_n \cup Y_n^+$.
6. $n \rightarrow n + 1$, repeat the steps 2. to 5. until $Y_n^+ = \emptyset$.

Example 2. Consider the system in Example 1. We will explain the refinement algorithm starting with a coarse equidistant grid $X_{N_1} = \{(x, y) \in \mathbb{R}^2 \mid x, y \in \{-h, 0, h\}\} \setminus \{(0, 0)\}$ with $h = 1$ and the set $K = [-1, 1]^2$, i.e. $N_1 = 8$.

- The first step of the algorithm $n = 1$: We start with an initial grid $X_1 = \{x_i\}_{i=1}^8$ and calculate the Lyapunov function v_1 on the grid X_1 . Figure 4 (a) shows the level set $v'_1(x, y) = 0$ with the area where $v'_1(x, y) > 0$ in green. We can also see the Voronoi diagram for our grid points and the circled Voronoi vertices which will all be added to the set X_1 , since they are all located in the green area. In this case, we have $Y_1^+ = Y_1$ and $Y_1^- = \emptyset$, i.e. all four new points (the equilibrium is not part of Y_1) have positive orbital derivative and will be added to the grid.



(a) The first step $n = 1$ with 8 grid points.

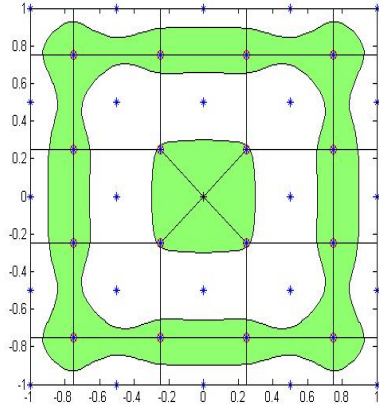
(b) The second step $n = 2$ with 12 grid points.

FIGURE 4. The first two steps, $n = 1, 2$, of the refinement algorithm. Both figures show the level set $v'(x, y) = 0$, which divides the region into areas with $v'(x, y) > 0$ (green), and areas with $v'(x, y) < 0$ (white). The grid points are blue * and the Voronoi diagrams with Voronoi vertices (black o) are shown.

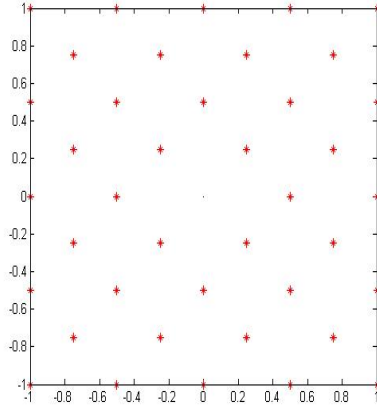
- The second step of the algorithm $n = 2$: Our new set of grid points is now $X_2 = X_1 \cup \{y_j\}_{j=1}^4$. Figure 4 (b) shows the level set $v'_2(x, y) = 0$. Again, we generate a Voronoi diagram for the set X_2 and determine the Voronoi vertices to be added to the existing grid points, see Figure 4 (b).

- The third step of the algorithm $n = 3$: The new set of grid points is $X_3 = X_2 \cup \{y_j\}_{j=1}^{12}$. After going through the same steps again we show in Figure 5 (a) the grid and level set $v'_3(x, y) = 0$.
- After the fourth step, the set $Y_4^+ = \emptyset$ and the algorithm terminates as all Voronoi vertices have negative orbital derivative. Figure 5 (b) shows the final set of grid points $X_4 = X_3 \cup \{y_j\}_{j=1}^{12}$ and no more areas with positive orbital derivative (green areas). Thus, we have found a Lyapunov function and will now be able to determine sublevel sets, which are subsets of the domain of attraction. We plot the graph of the Lyapunov function in Figure 6 (a) and its sublevel sets in Figure 6 (b).

To show that v' is negative, we have checked that the sign of $v'(x, y)$ is negative on the grid $X_{check} = \{(x, y) \mid x, y \in \{0, \pm h_{check}, \pm 2h_{check}, \dots, \pm 1\}\} \setminus \{(0, 0)\}$ with $h_{check} = 10^{-3}$.



(a) The third step of the refinement algorithm showing the level set $v'_3(x, y) = 0$, in green the areas where $v'_3(x, y) > 0$. The 24 grid points of X_3 (blue *) and Voronoi vertices (black o) to be added to set X_3 .

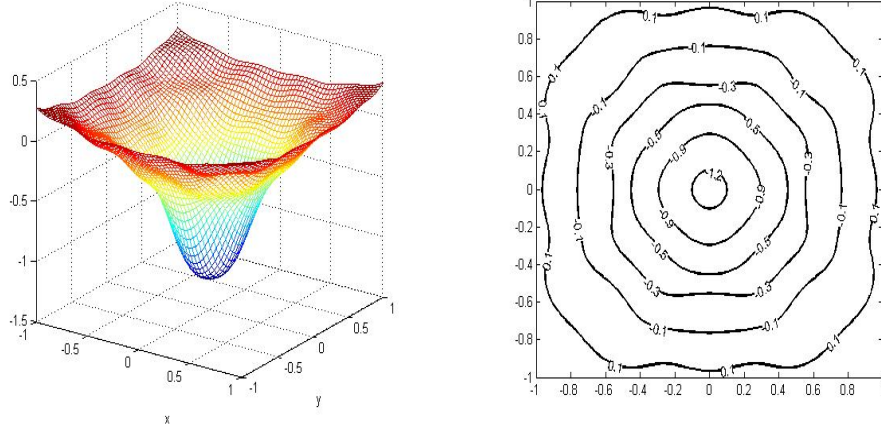


(b) The final set X_4 of 40 grid points (red *) after the refinement algorithm has terminated – no areas where $v'_4(x, y) > 0$ are left.

FIGURE 5. (a) shows the third step of the refinement algorithm, and (b) shows the final set of grid points with no areas of positive orbital derivative.

When solving the same example with a regular grid in Example 1, we needed $N = 168$ points, whereas with the refinement algorithm we have constructed a Lyapunov function with a grid of only $N_4 = 40$ points, hence we have reached our goal of having a dense enough grid with fewer points. The sublevel sets of the Lyapunov function with the refinement algorithm are similar to those we obtained with the regular grid $N = 168$, see Figures 1 (b) and Figure 6 (b).

After successfully testing the refinement algorithm on a linear system, we will now examine it on a nonlinear one.



(a) The constructed Lyapunov function $v_4(x, y)$.

(b) Different sublevel sets of $v_4(x, y)$.

FIGURE 6. (a) The constructed Lyapunov function $v_4(x, y)$ with the refinement algorithm (b) and its sublevel sets for different levels, which are all subsets of the domain of attraction.

Example 3. Consider the nonlinear system [10, Example 4.3]

$$\begin{cases} \dot{x} = -x - 2y + x^3, \\ \dot{y} = -y + \frac{1}{2}x^2y + x^3. \end{cases}$$

For this example, we approximate the Lyapunov function satisfying $V'(x) = -\|x\|^2$. We have used the Wendland function $\psi_{6,4}$ with $c=1$ and started with the grid $X_1 = \{(x, y) \in \mathbb{R}^2 \mid x, y \in \{0, \pm h, \dots, \pm 1\}\} \setminus \{(0, 0)\}$ with $h = 0.2$, i.e. $N_1 = 24$ points. Figure 7 (a) shows the region where $v'_1(x, y) > 0$ (green) and Figure 7 (b) shows the Voronoi diagram and vertices Y_1 . The red points marked with o (14 points) have positive orbital derivative and form the set Y_1^+ , these are the points that will be added to the previous grid.

Figure 8 (a) shows the level sets of the approximation v_2 , using the refined grid $X_2 = X_1 \cup Y_1^+$; the orbital derivative of v_2 for all points of X_2 is negative by construction.

After four refinement steps, the algorithm terminates and the final function satisfies $v'_4(x, y) < 0$ everywhere; no green areas (where $v'_4(x, y) > 0$) occur in Figure 8 (b). To show that v'_4 is negative, we have checked that the sign of $v'_4(x, y)$ is negative on the grid $X_{\text{check}} = \{(x, y) \mid x, y \in \{0, \pm h_{\text{check}}, \pm 2h_{\text{check}}, \dots, \pm 1\}\} \setminus [-0.1, 0.1]^2$ with $h_{\text{check}} = 10^{-3}$ – note that we have excluded a small neighborhood of the equilibrium because of the discussion in Remark 5.

Figure 9 (a) shows the graph of the constructed Lyapunov function $v_4(x, y)$, and Figure 9 (b) shows some of its sublevel sets. To construct a Lyapunov function with a regular grid which has negative orbital derivative on X_{check} , we need a grid of $N = 360$ points.

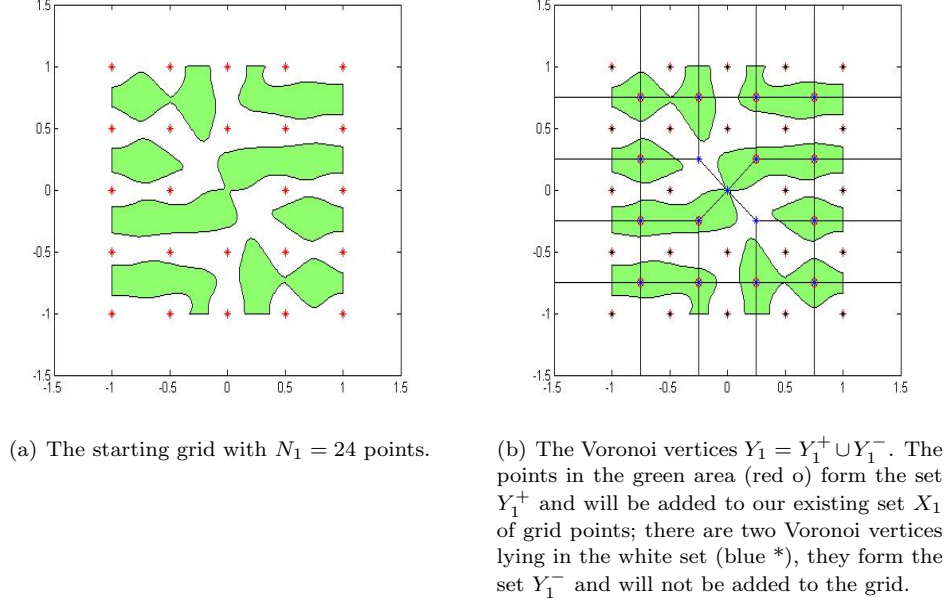


FIGURE 7. Both figures show the level set $v'_1(x, y) = 0$, which divides the region into areas with $v'_1(x, y) > 0$ (green), and areas with $v'_1(x, y) < 0$ (white). The grid points $N_1 = 24$ of the initial grid are red *.

Example 4 (Three-dimensional system). *Consider the 3-dimensional system given in [6, Example 6.4]*

$$\begin{cases} \dot{x} = x(x^2 + y^2 - 1) - y(z^2 + 1), \\ \dot{y} = y(x^2 + y^2 - 1) + x(z^2 + 1), \\ \dot{z} = 10z(z^2 - 1). \end{cases}$$

The system has an exponentially stable equilibrium at $(0, 0, 0)$ and its domain of attraction is given by

$$A(0, 0, 0) = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 < 1, |z| < 1\}.$$

In [6, Example 6.4], an RBF approximation with 137 points resulted in a large area near the equilibrium with positive orbital derivative; larger than the set $[-0.2, 0.2]^3$, which is later excluded in our example. With a modified algorithm, using the Taylor polynomial at the equilibrium, this was overcome in [6, Example 6.4].

As generally the domain of attraction is not known, we have chosen to use a grid in the set $K = [-0.9, 0.9]^3$, which is not a subset of the domain of attraction, but also does not include other invariant sets. This is a more realistic, but also more challenging test case for the method.

We choose the Wendland function $\psi_{6,4}$ with $c = 0.6$. We have started with a regular grid $X_N = \{(x, y, z) \in \mathbb{R}^3 \mid x, y, z \in \{0, \pm h, \pm 2h, \dots, \pm 0.9\}\} \setminus \{(0, 0, 0)\}$ and we have approximated the Lyapunov function $V = Q$ satisfying $Q' = -\|x\|^2$. In order to check the sign of the orbital derivative of the constructed Lyapunov

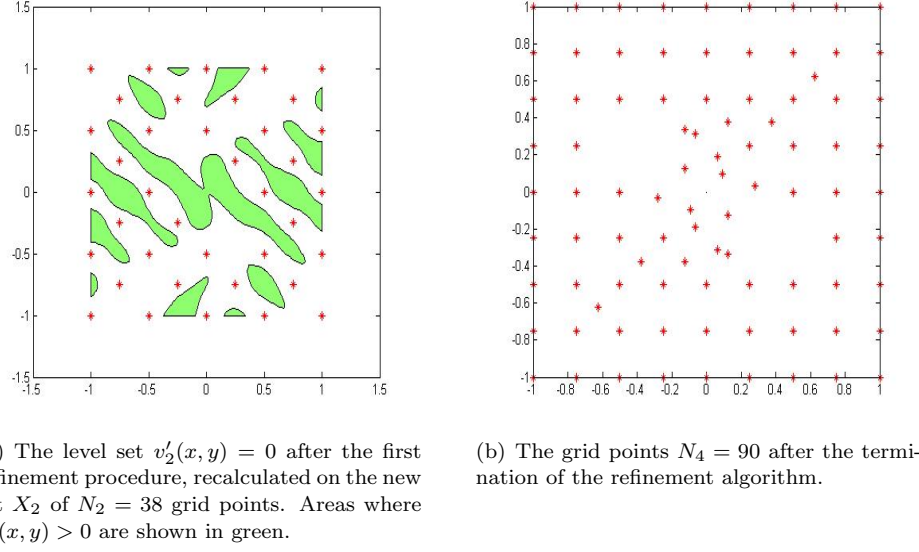


FIGURE 8. (a) The second step of the refinement algorithm and (b) the final step of the refinement algorithm: no areas with positive orbital derivative are left.

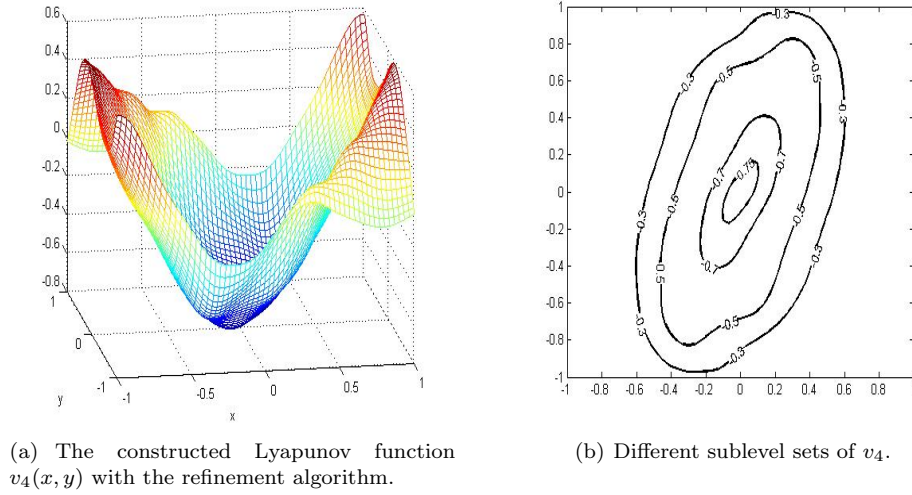
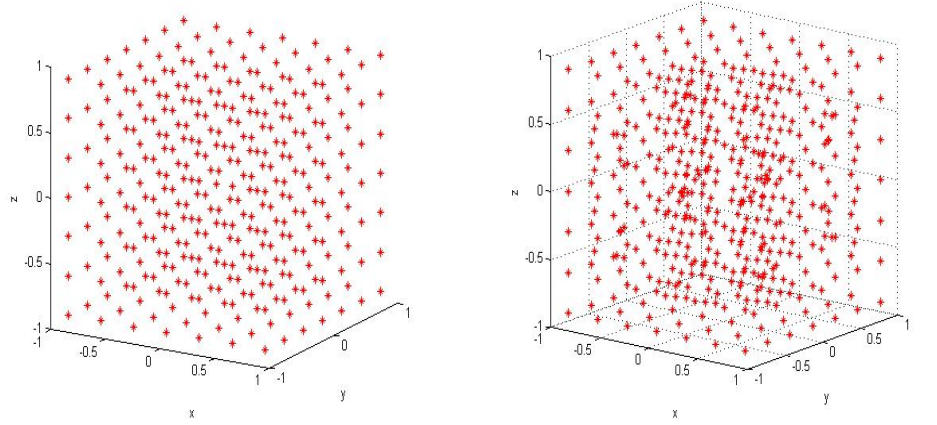


FIGURE 9. (a) The constructed Lyapunov function $v_4(x, y)$ with the refinement algorithm and (b) different sublevel sets of v_4 .

function v on finitely many points, we fix a grid $X_{check} = \{(x, y, z) \in \mathbb{R}^3 \mid x, y, z \in \{0, \pm h_{check}, \dots, \pm 0.9\}\} \setminus [-0.2, 0.2]^3$ with $h_{check} = 10^{-2}$.

To construct a Lyapunov function v with a regular grid which has negative orbital derivative on X_{check} we need $N = 2196$ points ($h = 0.15$).



(a) The initial grid X_1 with $N_1 = 342$ points. (b) The final grid X_2 with $N_4 = 458$ points.

FIGURE 10. (a) shows the distribution the initial grid points, (b) the distribution of final grid points after the last refinement step.

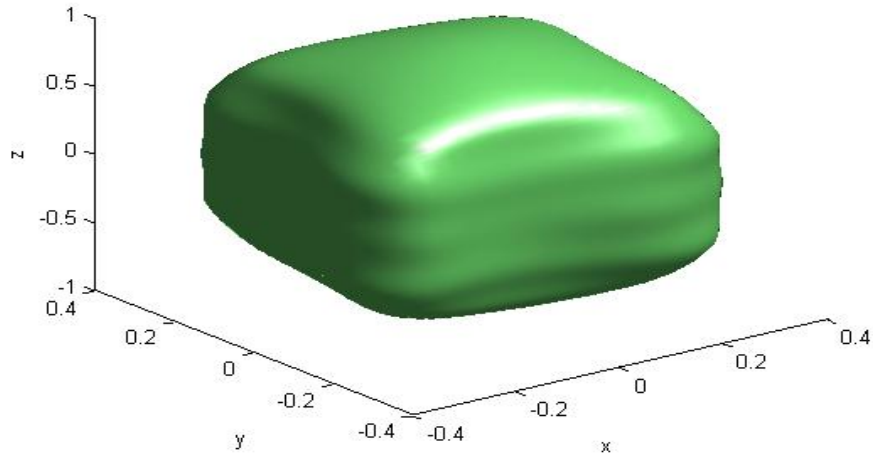


FIGURE 11. The figure shows a level set of the constructed Lyapunov function v_4 at value -1.2474 .

Now, applying our refinement algorithm with an initial set $N_1 = 342$ grid points, gives us a final set of grid points $N_4 = 458$, thus reducing again the number of points needed by a factor 4. The constructed Lyapunov function v has negative orbital derivative on the set X_{check} . Figure 10 (a) shows the initial grid X_1 , and Figure 10 (b) the grid X_4 after the refinement algorithm. Figure 11 displays a sublevel set of

the Lyapunov function v_4 , which is a subset of the domain of attraction, obtained at level value -1.2474 .

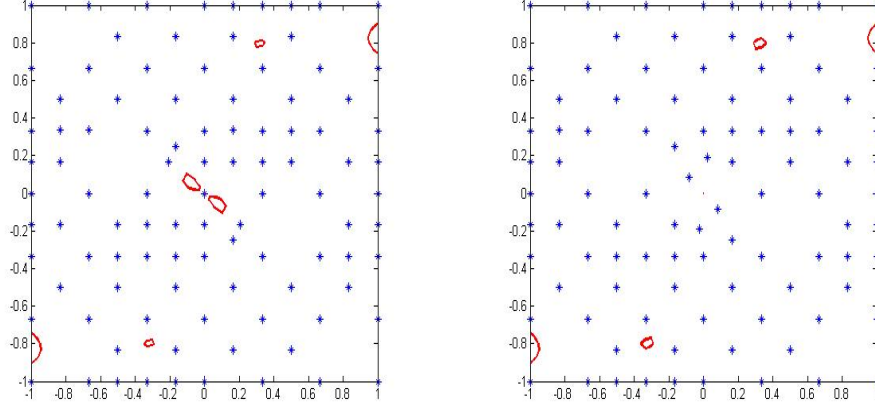
4.3. Starting grid. We have investigated how the starting grid influences the refinement algorithm.

4.3.1. First example. Considering again Example 3, we have started with regular grids of 16 to 360 points in $K = [-1, 1]^2$. In Table 2 we have listed the value of h for the initial grid and the corresponding number of points in the initial grid $N_{initial}$ as well as the number of points in the grid after the refinement algorithm has terminated. Moreover, it shows the running time for solving the linear systems in all refinement steps (time 1) and for calculating and plotting the orbital derivative of the constructed Lyapunov function for the final set of grid points (time 2), note that this time is proportional to the number N_{final} of grid points in the final grid. For all calculations we have used a standard laptop with an Intel(R) Core (TM) i5-3550 CPU @ 3.30 GHz processor.

h	$N_{initial}$	N_{final}	time 1 (refinement)	time 2 (plot)
2/3	16	99 (patches)	4.5 sec.	15.6 sec.
1/2	24	90	2.4 sec.	14.2 sec.
2/5	36	112	5 sec.	17.5 sec.
1/3	48	96 (patches)	4 sec.	15.3 sec.
1/4	80	88 (patches)	1.5 sec.	14 sec.
1/5	120	126 (patches)	3 sec.	20 sec.
1/6	168	176	9 sec.	27.5 sec.
1/7	224	224 (patches)	2.2 sec.	35 sec.
1/8	288	288 (patches)	4 sec.	45 sec.
1/9	360	360	6 sec.	1 min.

TABLE 2. The number of grid points and the value of h , we have used to construct Lyapunov functions for Example 3. This table shows the initial sets of regular grid points and the final number of points after refinement. Moreover, it shows the running time for solving the linear systems in all refinement steps (time 1) and for calculating and plotting the orbital derivative of the constructed Lyapunov function for the final set of grid points (time 2). “Patches” means that after termination of the refinement algorithm there are still areas with positive orbital derivative remaining. The shortest successful construction of a Lyapunov function is achieved with an initial grid of 24 points.

For small (16) and larger (48, 80, 120) numbers in our starting grid, the refinement algorithm stopped, but there were patches of areas where $v'(x, y) > 0$, see, for example, Figure 12. On the other hand, when we started with 24, 36 or 168 grid points, we achieved good results with no patches remaining at the end, see, for example, Figure 13. There were also some cases, for 224 and 288 starting points, where the refinement algorithm did not add any more points but we still had small patches. Finally, with 360 points, we reached the case where the refinement did not add any points and at the same time we did not have small patches of positive orbital derivatives in $[-1, 1]^2 \setminus [-0.1, 0.1]^2$.



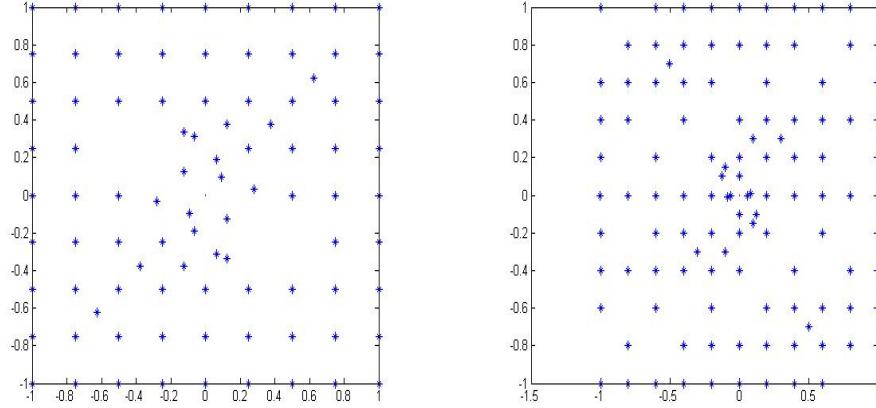
(a) The refinement with 16 initial grid points. (b) The refinement with 48 initial grid points.

FIGURE 12. (a) The level set $v'(x, y) = 0$ of the orbital derivative after the last refinement step started with 16 points and ended up with 99 points, (b) the level set $v'(x, y) = 0$ of the last refinement step started with 48 points and ended up with 96 points. In both cases we can see the small patches remaining at the end of the refinement procedure, where the orbital derivative is positive (red areas).

The reason for these patches is that the last refinement step placed all Voronoi vertices in areas where $v'(x, y) < 0$, thus missing the areas where refinement would be necessary. Thus, a further test to ensure that the orbital derivative is negative is necessary; therefore, we have checked the sign of v' on the grid $X_{check} = \{(x, y) \in \mathbb{R}^2 \mid x, y \in \{0, \pm h_{check}, \dots, \pm 1\}\} \setminus [-0.1, 0.1]^2$ with $h_{check} = 10^{-3}$. Moreover, future work will include an improvement of the refinement algorithm to add points to the grid in these areas.

The refinement algorithm solves (small) linear systems in each refinement step, while for the grid with 360 points only one (larger) linear system needs to be solved (time 1). Comparing the four successful constructions of a Lyapunov function (24, 36, 168 and 360 initial points), the starting grids with 24 and 36 have a shorter time 1 than the grid with 360 points, where no refinement step is necessary, so here solving several small systems is shorter than solving a large one. Starting with 168 points, however, takes longer. However, as the number of points in the final grid is proportional to the time to calculate and plot the orbital derivative (time 2), which takes much longer than solving the linear systems, all smaller starting grids (24, 36, 168) take a shorter overall time (time 1+ time 2).

4.3.2. Second example. A similar behaviour was observed in another example, namely [6, Example 2.10]. In this example, we have used the Wendland function $\psi_{6,4}$ with $c = 1$ and approximated the Lyapunov function satisfying $T' = -1$ starting with regular grids of 16 to 168 points. However, in this example we have distributed our points on $K = [-0.9, 0.9]^2$. The domain of attraction of this system is given by



(a) The refinement with 24 initial grid points. (b) The refinement with 36 initial grid points.

FIGURE 13. The final grid points after the last refinement step, (a) started with 24 points and ended up with 90 points, (b) started with 36 points and ended up with 112 points. As we can see, in both cases there are no small patches remaining at the end of the refinement procedure.

h	$N_{initial}$	N_{final}	time 1 (refinement)	time 2 (plot)
0.6	16	60 (patches)	0.6 sec.	9 sec.
0.45	24	104	3 sec.	15.5 sec.
0.36	36	88	1.8 sec.	13 sec.
0.3	48	60 (patches)	0.4 sec.	9 sec.
0.225	80	104	1.7 sec.	15.5 sec.
0.18	120	132	4.5 sec.	19.5 sec.
0.15	168	168	1.2 sec.	25 sec.

TABLE 3. The number of grid points and the values of h we have used to construct Lyapunov functions for Example 2.10 of [6]. This table shows the initial sets of regular grid points and the final number of points after refinement. Moreover, it shows the running time for solving the linear systems in all refinement steps (time 1) and for calculating and plotting the orbital derivative of the constructed Lyapunov function for the final set of grid points (time 2). The shortest successful construction of a Lyapunov function is achieved with an initial grid of 168 points, without any refinement step (time 1); the shortest construction and plot (time 1 + time 2), however, is achieved with an initial grid of 36 points.

$A(0,0) = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}$, so K actually is not a subset of the domain of attraction. As usually the domain of attraction is not known in advance, this is a realistic situation, and even in this situation, the refinement algorithm works well, see Table 3. We used $X_{check} = \{(x,y) \in \mathbb{R}^2 \mid x,y \in \{0, \pm h_{check}, \dots, \pm 0.9\}\} \setminus [-0.1, 0.1]^2$

with $h_{check} = 10^{-3}$ to check that the sign of v' is negative for the examples with 24, 36, 80, 120 and 168 starting points. In this example, we reached the case where the refinement algorithm did not add more points and we did not have patches remaining, with 168 regular grid points, see Table 3.

Although in this example, the shortest time to solve the linear system(s) is achieved with the grid of 168 points, without any refinement, again the sum of time 1 and time 2 (solving linear system(s) and plotting the orbital derivative) is the longest for the initial grid of 168 points.

4.3.3. *Third example.* The influence of the starting grid on the refinement algorithm was also investigated in the 3-dimensional system introduced in Example 4. In Table 4, we have presented the values of h and the corresponding number of regular starting grid points, the final number of grid points after the refinement process and the running time in each case.

h	$N_{initial}$	N_{final}	time 1 (refinement)	time 2 (plot)
0.45	124	180 (patches)	5 sec.	25 min.
0.3	342	458	50 sec.	1 hr. 15 min.
0.225	728	728 (patches)	23.5 sec.	2 hrs.
0.18	1330	1330 (patches)	1 min. 20 sec.	3 hrs. 30 min.
0.15	2196	2196	4 min.	6 hrs. 47 min.

TABLE 4. The number of grid points and the values of h we have used to construct Lyapunov functions for Example 4. The table shows the number of regular grid points we started with and the final number of points after refinement. Moreover, it shows the running time for solving the linear systems in all refinement steps (time 1) and for calculating and plotting the orbital derivative of the constructed Lyapunov function for the final set of grid points (time 2). The shortest successful construction of a Lyapunov function is achieved with an initial grid of 342 points.

Again, when we started with a coarse grid (124 initial points), the refinement algorithm stopped but with some remaining patches where $v'(x, y, z) > 0$, see Figure 14 (a). Moreover, starting with a finer grid of 728 or 1330 initial points, the refinement algorithm did not add any more points but we still have patches, see Figure 14 (b) and (c). However, starting with a regular grid of 342 points, and after 4 refinement steps, we ended up with a desirable result with no patches remaining at the end. A similar result was achieved with a regular fine grid of 2196 points, except for a small neighborhood of the origin, see Figure 15. Recall that, for this example, we used $X_{check} = \{(x, y, z) \in \mathbb{R}^3 \mid x, y, z \in [0, \pm h_{check}, \dots, \pm 0.9]\} \setminus [-0.2, 0.2]^3$ with $h_{check} = 10^{-2}$, to check the sign of v' is negative everywhere for 342 and 2196 points in the initial grid.

Here, there is a considerable difference between the two successful constructions of more than a factor four, both in the times to solve the linear systems (time 1) and in the time to plot the orbital derivative (time 2).

Summarising, starting with a grid which is too coarse or too fine may end the refinement algorithm with a function that has still areas where the orbital derivative

is positive. A moderately coarse grid is the most promising starting point. In most cases, the time to calculate a Lyapunov function was shorter by using the refinement algorithm, even if it involves solving a system of linear equations in each refinement step (time 1). An even greater advantage of the refinement algorithm, however, becomes apparent when using the calculated Lyapunov function further, e.g., to calculate and plot its orbital derivative (time 2), which is proportional to the number of points in the final grid; here, the reduction in the number of grid points pays off considerably.

5. Conclusion and outlook. In this paper we have introduced a refinement algorithm for the Radial Basis Function method to construct Lyapunov functions. The recursive refinement of the grid is done by considering the Voronoi vertices of the previous grid as possible new grid points, which are added if the orbital derivative of the previous approximation is non-negative. The algorithm terminates, if no points are added.

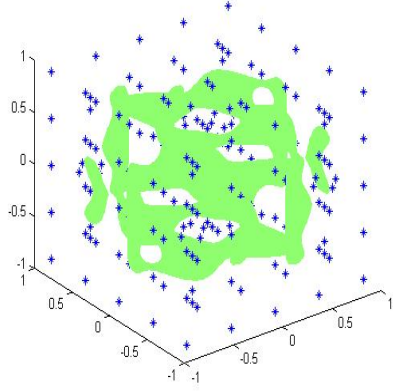
Compared to using a regular grid, the refinement algorithm is able to reduce the required number of grid points by nearly a factor of 4 in some examples and a factor of 2 in others, depending on the system we are solving and which Lyapunov function we approximate (i.e., T' or Q'). This reduces the computational effort and time considerably when evaluating the constructed Lyapunov function. It even succeeded in constructing a Lyapunov function if the grid was placed in a set, which was not a subset of the domain of attraction.

When the refinement algorithm terminates, since all Voronoi vertices have already negative orbital derivative, there may still be points in between, where the orbital derivative is positive. Hence, the values of $v'(x)$ have been checked to be negative on a very fine grid. Future work will improve this by providing a reliable verification of the negativity of the orbital derivative using error estimates. This can then be used for a modified refinement algorithm, where points in these areas of positive orbital derivative will be added, and the refinement continues until a Lyapunov function is found. Further work on this refinement algorithm could also include the derivation of optimal initial grid configurations.

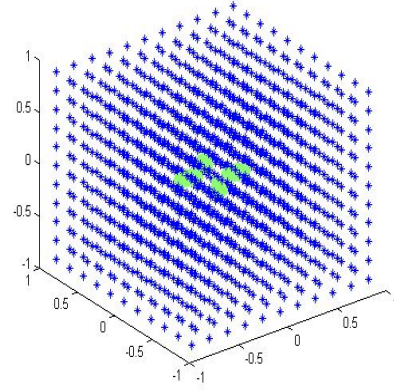
This paper has established the first refinement algorithm for the construction of Lyapunov functions using Radial Basis Functions. It is a considerable improvement from the regular grid, that was used until now, and provides a systematic way to reduce the number of grid points and tackle larger problems in higher dimensions.

REFERENCES

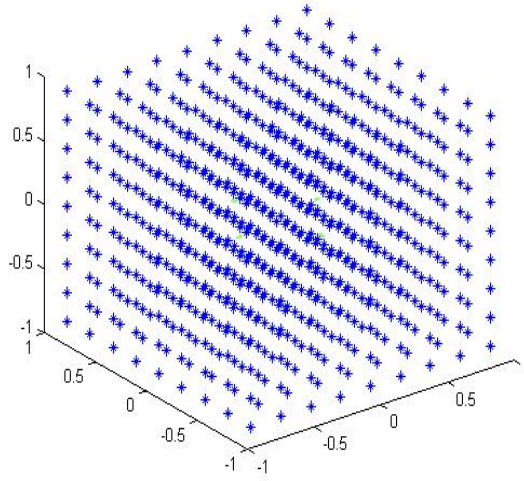
- [1] M. Berg, O. Cheong, M. Kerveld, and M. Overmars. *Computational geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2008.
- [2] M. D. Buhmann. Radial basis functions. In *Acta numerica, 2000*, volume 9 of *Acta Numer.*, pages 1–38. Cambridge Univ. Press, Cambridge, 2000.
- [3] F. Camilli, L. Grüne, and F. Wirth. A generalization of Zubov’s method to perturbed systems. *SIAM J. Control Optim.*, 40(2):496–515, 2001.
- [4] M. Dellnitz and O. Junge. Set oriented numerical methods for dynamical systems. In *Handbook of dynamical systems, Vol. 2*, pages 221–264. North-Holland, Amsterdam, 2002.
- [5] M. Floater and A. Iske. Multistep scattered data interpolation using compactly supported Radial Basis Functions. *J. Comput. Appl. Math.*, 73(1-2):65–78, 1996.
- [6] P. Giesl. *Construction of Global Lyapunov Functions Using Radial Basis Functions*. Lecture Notes in Math. 1904, Springer, 2007.
- [7] P. Giesl. Construction of a local and global Lyapunov function using Radial Basis Functions. *IMA J. Appl. Math.*, 73(5):782–802, 2008.



(a) The refinement with 124 initial grid points.



(b) The refinement with 1330 initial grid points.



(c) The refinement with 728 initial grid points (4 little green patches).

FIGURE 14. (a) The level set $v'(x, y, z) = 0$ of the orbital derivative after the last refinement step started with 124 points and ended up with 180 points, (b) and (c) the level set $v'(x, y, z) = 0$ of the orbital derivative calculated with 728 and 1330 points, where the refinement did not add more points. In all cases we can see some patches remaining at the end (green), where $v'(x, y, z) > 0$.

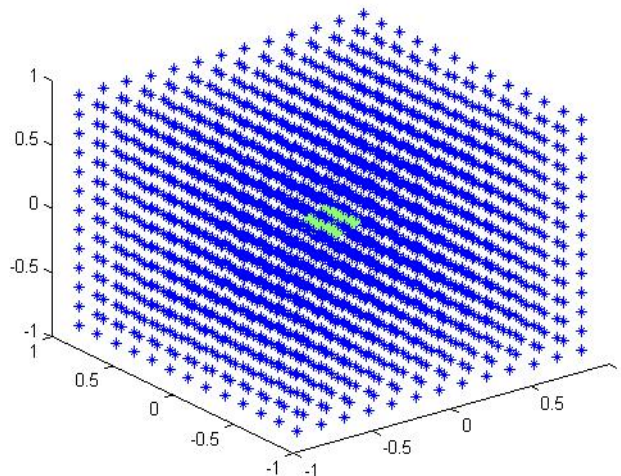


FIGURE 15. The level set $v'(x, y, z) = 0$ of the orbital derivative calculated with a regular grid of 2196 points. As we can see there are no patches remaining at the end except for a small neighbourhood, $[-0.2, 0.2]^3$, of the equilibrium point $(0,0,0)$.

- [9] P. Giesl and S. Hafstein. Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems - Series B*, submitted.
- [10] P. Giesl and H. Wendland. Meshless collocation: error estimates with application to Dynamical Systems. *SIAM J. Numer. Anal.*, 45(4):1723–1741, 2007.
- [11] L. Grüne. An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation. *Numer. Math.*, 75(3):319–337, 1997.
- [12] L. Grüne. *Asymptotic behavior of dynamical and control systems under perturbation and discretization*, volume 1783 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2002.
- [13] S. Hafstein. A constructive converse Lyapunov theorem on exponential stability. *Discrete and Continuous Dynamical Systems - Series A*, 10(3):657–678, 2004.
- [14] S. Hafstein. *An algorithm for constructing Lyapunov functions*. Monograph. Electron. J. Diff. Eqns., 2007.
- [15] C. S. Hsu. *Cell-to-cell mapping*, volume 64 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1987.
- [16] A. Iske. On the construction of kernel-based adaptive particle methods in numerical flow simulation. In *Recent developments in the numerics of nonlinear hyperbolic conservation laws*, volume 120 of *Notes Numer. Fluid Mech. Multidiscip. Des.*, pages 197–221. Springer, Heidelberg, 2013.
- [17] S. Iyengar, K. Boroojeni, and N. Balakrishnan. *Mathematical Theories of Distributed Sensor Networks*. Springer, New York, 2014.
- [18] Z. Jian. *Development of Strong Form Methods with Applications in Computational Mechanics*. PhD thesis: National University of Singapore, Singapore, 2008.
- [19] C. Kellett. Converse Theorems in Lyapunov’s Second Method. *Discrete and Continuous Dynamical Systems - Series B*, submitted.
- [20] R. Klein. *Concrete and Abstract Voronoi Diagrams*. Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1989.
- [21] J. Massera. On Liapounoff’s conditions of stability. *Ann. of Math.*, 50(2):705–721, 1949.
- [22] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Pranja, P. Seiler, and P. Parrilo. *SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB*. User’s guide. Version 3.00 edition, 2013.

- [23] P. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis: California Institute of Technology Pasadena, California, 2000.
 - [24] F. Preparata and M. Shamos. *Computational geometry*. Texts and Monographs in Computer Science. Springer-Verlag, New York, 1985.
 - [25] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Approx. Theory*, 18(3):548–585, 1995.
 - [26] R. Sibson. Development of strong form methods with applications in computational mechanics. In V. Barnett, editor, *Interpolating Multivariate data*, chapter 2. John Wiley and Sons, New York, 1981.
 - [27] H. Wendland. Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree. *J. Approx. Theory*, 93:258–272, 1998.
 - [28] H. Wendland. *Scattered data approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2005.
 - [29] X. Zhang, R. Ding, and Y. Li. Adaptive RPIM meshless method. In *Proceedings of the 2011 International Conference on Multimedia Technology (ICMT)*, pages 2388–2392. IEEE, 2011.
- E-mail address:* nm230@sussex.ac.uk, p.a.giesl@sussex.ac.uk